XAR-Miner: Efficient Association Rules Mining for XML Data

Ji Zhang
Dept. of Computer Science
University of Toronto
Toronto, Canada, M5S3G4
jzhang@cs.toronto.edu

Han Liu
Dept. of Computer Science
University of Toronto
Toronto, Canada, M5S3G4
hanliu@cs.toronto.edu

Wei Wang
Nanjing Normal University,
Nanjing, China
wwang@pami.uwaterloo.ca

ABSTRACT

In this paper, we propose a framework, called XAR-Miner, for mining ARs from XML documents efficiently. In XAR-Miner, raw data in the XML document are first preprocessed to transform to either an Indexed Content Tree (IX-tree) or Multi-relational databases (Multi-DB), depending on the size of XML document and memory constraint of the system, for efficient data selection and AR mining. Task-relevant concepts are generalized to produce generalized meta-patterns, based on which the large ARs that meet the support and confidence levels are generated. The experiments conducted show that XAR-Miner is more efficient in performing a large number of AR mining tasks from XML documents than the state-of-the-art method of repetitively scanning through XML documents in order to perform each of the mining tasks.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *Data mining*.

General Terms

Management.

Keywords

Association Rule Mining, XML Data, Meta-Patterns, Concept Generalization.

1. INTRODUCTION

The fast-growing amount of XML-based information on the web has made it desirable to develop new techniques to discover patterns and knowledge from XML data. Association Rule (AR) mining is frequently used to reveal interesting trends, patterns, and rules in large datasets. Though we have witnessed intensive research work in AR mining in the past years, there has been very little work in the domain of AR mining from XML documents. The work in [1] uses the MINE RULE operator introduced in [2] for AR mining purposes in native XML documents. The Predicative Model Markup Language (PMML) is proposed to present various patterns such as association rules and decision trees extracted from XML documents [3]. [4] presents a XML-enable AR mining framework, but does not give any details on how to implement this framework efficiently. [5] claims that XML AR can be simply accomplished using XQuery language. The major problems with the state-of-the-art methods are two-fold: (i) these approaches select data from native XML document, thus the efficiency of these approaches is low because of the normally huge volume of XML data that need to be scanned in AR mining, and (ii) They lack the mechanism to control the degree to which generalization is performed. Under-generalization or over-generalization may seriously affect the effectiveness of the AR mining.

To address the above problems, we propose a new framework, called XAR-Miner, to efficiently mine ARs from XML documents. In XAR-Miner, XML data are extracted and organized in a way that is suitable for efficient data selection in the AR mining. Concepts relevant to the AR mining task are generalized, if necessary, to a proper degree in order to generate meaningful yet nontrivial ARs.

2.THE FRAMEWORK

The framework of AR mining of XML data consists of the following major parts: (1) Pre-processing (i.e., construction of the Indexed XML Tree (IX-tree) or Multiple Relational Databases (Multi-DB)); (2) Generation of generalized meta-patterns; and (3) Generation of large ARs of generalized meta-patterns. The overview of the framework is shown in Figure 1.

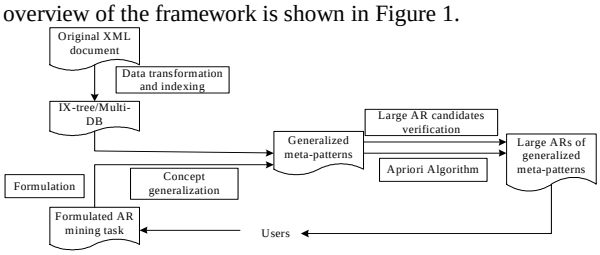

Figure. 1. The overview of XAR-Miner

2.1 XML Data Extraction and Transformation

The preprocessing work of XAR-Miner is to extract information from the original XML document and transform them into a way that is suitable for efficient AR mining. Specifically, we build *Indexed XML Tree* (IX-tree) when all the XML data can be loaded into main memory, and build *Multi-relational Databases* (Multi-DB) otherwise. We will evaluate the size of the XML data involved and the main memory available to select the proper strategy for XML data transformation and storage before AR mining tasks are preformed.

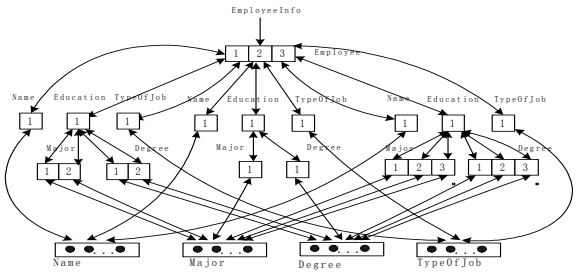

Figure 2. IX-tree of EmployeeInfo.xml

An *Inde* ɔe IX-tree=<*V*, *E*, *A*>, where *V* is the vertex set, *E* is the edge set and *A* is the indexed array set. *V* is set of nodes appearing in the XML document. The intermediate nodes in the IX-tree store the

addresses of its immediate parent and children. An edge $e(v_1,v_2)$ in the IX-tree connects the two vertices v_1 and v_2 using a bi-directional link. The set of indexed arrays A positioned at the bottom level within the IX-tree stores the data in the leaf element or attribute nodes in the original XML document. The IX-tree of a sample XML document (Employeeinfo.xml) is shown in Figure 2.

Under the circumstance that the size of XML data exceeds the main memory available, we alternatively choose to construct Multi-relational Databases to accommodate the extracted XML data. In this architecture, an XML document will be transformed to a few relational databases, each of which will store the indexed structural values of a leaf node in the XML document. We use the notion of Serial Xpath String (SXS) to identify each XML data in relational databases. The Serial Xpath String of an XML data x is a string that gives the ordinal numbers of concepts of different levels along the path from the root to the leaf node of x. Any SXS will start with the root node and the ordinal number of concepts along the path will be delimited by dashed lines. Figure 3 shows the Multi-DB of the same sample XML document.

Name	SXS
James Wang	R-1-1
Ghai Vandana	R-2-1
Linda Lee	R-3-1

TypeOfJob	SXS
System Analyst	R-1-1
Accountant	R-2-1
Project Manager	R-3-1

Major	SXS
Computer Science	R-1-1-1
Computer Science	R-1-1-2
Accounting	R-2-1-1
Mathematics	R-3-1-1
Management	R-3-1-2
Economics	R-3-1-3

Degree	SXS
B.S	R-1-1-1
M.Sc	R-1-1-2
B.A	R-2-1-1
B.S	R-3-1-1
M.B.A	R-3-1-2

Figure 3. Multi-relational database transformation of Employeeinfo.xml

1.2 Data Selection from IX-tree and Multi-DB

The paths between the instances of related concepts are needed for retrieving data from IX-tree or Multi-DB. In IX-tree, the bi-directional linking between parent and child nodes in the tree hierarchy realizes fast top-down and bottom-up traversal, which facilitates the retrieval of the value of the concepts involved in AR mining. To create the path connecting related concepts, the Nearest Common Ancestor Node (NCAN) of these concepts must first be found. The NCAN of elements $e_1,e_2,\ldots,e_n$ in the hierarchy of an XML document H is defined as the common ancestor node in H that is the closest to e_i ($1 \leq i \leq n$). In Multi-DB architecture, SXS for each XML data and Xpath for each relational database are created during data transformation that perfectly maintains the hierarchical information of concepts in the original XML document. The related XML data have the identical substring of varied length in their SXSs. This identical substring is the Xpath from the root to the NCAN of these related concepts. The ordinal number of the NCAN of the concepts can be used to identify data uniquely. This observation can help to retrieve the values/instances of related concepts easily.

1.3 Generate Generalized Meta-Patterns

The task-relevant raw XML will usually be generalized in order to generate ARs that are significant enough, which is necessitated by the sparsity of the XML data involved. Data should be generalized properly in order to find significant yet nontrivial ARs. Under-generalization may not render the data dense enough for finding patterns extracting significant ARs. However, over-generalization may lead to patterns that extract trivial ARs that are not depended on the support and confidence thresholds. In our work, we utilize a metric to measure the degree of generalization and two constraints, *Min_gen* and *Max_gen*, are used to help avoid under-generalization or over-generalization. In addition, a *Generalization Reference Table* (GRT) is created offline to provide guidance regarding the concept generalization.

1.4 Generate Large AR Rules

After the generalized meta-patterns have been obtained, XAR-Miner will generalize the raw XML data based on the meta-patterns and generate large ARs w.r.t. the user-specified minimum support (*minsup*) and confidence (*minconf*) requirements using Apriori algorithm.

It is easy to know that if R_2 is the generalized AR of R_1, then Support(R_1)≤Support(R_2). We can therefore infer that if R_2 is not a large AR, then R_1 is not a large AR either. This observation helps us devise an efficient algorithm to perform AR generation of generalized meta-patterns, which adopts a top-down strategy to traverse the generalization meta-patterns. The basic idea is that, instead of directly working on the raw generalized data to generate large ARs for a certain meta-pattern, the large ARs of its higher generalized meta-pattern are used to obtain the large AR candidates whose largeness can be easily verified. Obviously, verifying the largeness of ARs is much cheaper than mining all the large AR directly from the data.

3.CONCLUSIONS

We propose a framework, called XAR-Miner, to mine ARs from XML documents efficiently. XAR-Miner transforms data in the XML document and constructs an Indexed XML Tree (IX-tree) if the XML data can be fully loaded into main memory or Multi-relational databases (Multi-DB) otherwise that perfectly maintain the hierarchical information of XML data and perform indexing of the data to realize efficient retrieval of data in AR mining. Concepts that are relevant to the AR mining task are generalized to produce generalized meta-patterns. A suitable quantitative metric is devised for measuring the degree of concept generalization in order to prevent under-generalization or over-generalization. Resultant generalized meta-patterns are used to generate large ARs that meet the support and confidence levels.

4. REFERENCES

[1] D. Braga, A. Campi, M. Klemettinen and P. Lanzi. Mining Association Rules from XML Data, in Proceedings of *DaWaK'02*, pp. 21-30, Aix-en-Provence, France, 2002.

[2] R. Meo, G. Psaila and S. Ceri. A New Operator for Mining Association Rules, in Proceeding of *VLDB'96*, pp. 122-133, Bombay, India, 1996.

[3] PMML 2.0: Predicative Model Makeup Language.

[4] L. Feng, T. S. Dillon, H. Weigand, E. Chang. An XML-Enabled Association Rule Framework. In Proceedings of *DEXA'03*, pp 88-97, Prague, Czech Republic, 2003.

[5] W. W. Wan, G. Dobbie. Extracting association rules from XML documents using XQuery. In Proceedings of *WIDM'03*, pp. 94-97, New Orleans, Louisiana, USA, 2003.